

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Michal Chylík

Bakalářská práce

Vedoucí práce: Ing. Tomáš Fabián, Ph.D.

Ostrava, 2021

Abstrakt

Cieľom tejto bakalárskej práce je popísať čitateľovi priebeh odbornej praxe, ktorú som absolvoval vo firme ISSA CZECH s.r.o v pozícií PHP programátora a vývojára webových aplikácií. Náplňou praxe bol vývoj interného informačného systému pre túto firmu pomocou vlastného firemného frameworku v programovacích jazykoch najmä PHP a HTML. Práca sa zaoberá podrobným popisom riešených úloh a zadaní, ktoré som od firmy pravidelne dostával pri vykonávaní odbornej praxe. Ďalej práca obsahuje predstavenie spoločnosti, popis používaných jazykov a technológií s ktorými som pri vývoji pracoval, a taktiež zhodnotenie znalostí a skúseností ktoré mi škola a odborná prax priniesla.

Klíčové slová

ISSA CZECH s.r.o, individuálna odborná prax, framework, vývoj systému, webová aplikácia, Databáza MariaDB, PHP, HTML

Abstract

The aim of this bachelor's thesis is describe to the reader the course of professional practice, which I completed in the company ISSA CZECH s.r.o. in position of PHP programmer and web application developer. The content of the practice was the development of an internal information system for this company using its own company framework in programming languages, especially PHP and HTML. The work deals with a detailed description of solved tasks and assignments, which I regularly received from the company during the practice. Furthermore, the work contains a presentation of the company, a description of the languages and technologies used, which I worked with in the development, as well as an evaluation of the knowledge and experience that the school and professional practice brought me.

Keywords

ISSA CZECH s.r.o, individual professional practice, framework, development of system, web application, Database MariaDB, PHP, HTML

Podakovanie

Rád by som poďakoval celej spoločnosti ISSA CZECH s.r.o za umožnenie vykonania odbornej praxe a hlavne Ing. Lukášovi Novákovi za jeho trpezlivosť a ochotu vysvetľovať pri riešení úloh a zadaní. Podakovanie patrí aj Ing. Tomášovi Fabiánovi, Ph.D., ktorý ma viedol pri tvorbe tejto bakalárskej práce.

Obsah

Zoznam použitých symbolov a skratiek	6
Zoznam obrázkov	7
Zoznam tabuliek	8
Zoznam výpisov zdrojového kódu	9
1 Úvod	10
2 Predstavenie spoločnosti	12
2.1 Používané technológie vo firme	12
2.2 Produkty a služby	13
2.3 Popis pracovnej pozície	14
3 Špecifikácia zadania	15
3.1 Firemný Intranet	15
3.2 Hlavný cieľ praxe	16
4 Riešenie zadaných úloh	19
4.1 Používané jazyky a technológie	19
4.2 Oboznámenie sa s frameworkom	22
4.3 Katalógy	23
4.4 Pracovníci	25
4.5 Dochádzky	26
4.6 Stravenky	29
4.7 Moje hodnotenie	32
5 Závěr	35
Literatúra	36

Prílohy	37
A Dochádzky súhrn	38

Zoznam použitých skratiek a symbolov

AJAX	– Asynchronous JavaScript + XML
API	– Application Programming Interface
CSS	– Cascading Style Sheet
CSV	– Comma-separated values
GIS	– Geographical Information System
HTML	– Hypertext Markup Language
IKT	– Informačno-Komunikačné Technológie
JSON	– JavaScript Object Notation
LAN	– Local Area Network
Mbps	– Megabits per second
PDF	– Portable Document Format
PHP	– Hypertext Preprocessor
WAN	– Wide Area Network
XML	– eXtensible Markup Language

Zoznam obrázkov

3.1	Intranet - stará verzia	16
4.1	Dochádzky jednotlivca	27
4.2	Štatistika jednotlivca	28
4.3	Formulár modulu stravenky	30
4.4	Návrh databázy intranet	31

Zoznam tabuliek

4.1	Časová náročnosť zadaných úloh	32
-----	--	----

Zoznam výpisov zdrojového kódu

4.1	Funkcia generovania sviatkov	24
4.2	Relácie tabuľky pracovníci	25
4.3	Štruktúra tabuľky stravenky	29

Kapitola 1

Úvod

Táto bakalárska práca popisuje celý priebeh odbornej praxe, ktorú som vykonával počas 3. ročníka bakalárskeho štúdia na Fakulte elektrotechniky a informatiky VŠB-TUO. Prax som absolvoval v spoločnosti ISSA CZECH, ktorá sa zaoberá prácou s rôznymi informačnými technológiami. V tejto práci popisujem celý priebeh tejto praxe od zadaných úloh a ich riešení až po jej zhodnotenie.

Odbornú prax som si zvolil najmä z dôvodu, že som chcel získať znalosti a skúsenosti a podieľať sa na nejakom projekte, ktorý bude ďalej vystavený každodennému použitiu. Doteraz som skúsenosti získaval najmä pri školských úlohách a projektoch alebo v rámci samoštúdia v oblasti informačných technológií, ktorá ma najviac zaujímala. Tu som nadobudol dostatok skúseností na to, aby som sa mohol konečne dozvedieť ako prebieha vývoj reálneho projektu a zároveň sa na ňom podieľať. V podstate som sa chcel zoznámiť s pracovným životom v nejakej spoločnosti a získať prehľad o tom, čo by mohlo nasledovať po mojom štúdiu na vysokej škole.

Konkrétnu firmu pre bakalársku prax som hľadal na zozname odborných praxí v systéme Katis, kde sú firmy so záujmom prijať študentov do svojho kolektívu. Nachádza sa tu mnoho firiem so širokým spektrom zameraní v oblasti informačných technológií. Zameranie spoločnosti bolo jedným z najdôležitejších aspektov môjho výberu, čo v mojom prípade predstavovalo vývoj rozsiahlejšej webovej aplikácie v rámci frontendu alebo backendu. Vyhľadával som menšiu firmu s priateľským kolektívom, dobrým menom a prístupnou lokalitou.

Napokon som oslovil viacero firiem, no v žiadnej nebola komunikácia na takej úrovni ako v spoločnosti ISSA, ktorá vo veľkom rozsahu splnila moje očakávania. Veľmi prijateľný bol prístup firmy a jej kolektívu, ktorý od študenta neočakával znalosti programovania a práce s rôznymi nástrojmi a technológiami na pokročilej úrovni.

V druhej kapitole tejto práce popisujem vybranú spoločnosť, čím sa zaoberá, aké služby a produkty ponúka svoji zákazníkom, aké technológie v rámci IKT využíva a taktiež opisujem pracovnú pozíciu, ktorá mi bola po nástupe do firmy pridelená.

Následne v tretej kapitole oboznamujem čitateľa o tom aké zadanie som od vedea firmy dostal a čo bolo mojím finálnym cieľom pri vykonávaní praxe. Ďalšia kapitola obsahuje postup riešenia

všetkých zadaných úloh a stručný popis technológií, s ktorými som sa musel zoznámiť. Pri riešení zložitejších zadaní som popisoval detailný pracovný postup a v niektorých prípadoch aj samotné funkcie programu. Ďalej popisujem firemnú databázu, s ktorou som pracoval a časovú zložitosť vypracovania celého projektu, a môj osobný pohľad na splnené zadanie.

V závere tejto práce sa venujem popisu všetkých znalostí, ktoré som získal počas štúdia na vysokej škole ale aj znalosti, ktoré mi pri riešení úloh chýbali a musel som ich dopĺňať samoštúdiom.

Počas praxe som pracoval na vývoji informačného systému pre správu dát v spoločnosti. Táto práca zahŕňala inováciu intranetu zo starej verzie do novej s rozšírenou funkcionalitou a intuitívnym používateľským rozhraním s moderným dizajnom. Mojou najväčšou oporou bol firemný PHP framework, pomocou ktorého som si musel poradiť s riešením tohto komplexného systému na všetkých jeho vrstvách od tvorby tabuliek v databáze až po štylovanie užívateľského rozhrania.

Kapitola 2

Predstavenie spoločnosti

Česká spoločnosť ISSA CZECH s.r.o. bola založená v roku 1997 v Ostrave v období prudkého rozvoja informačných a komunikačných technológií v oblasti Internetu. Názov spoločnosti ISSA predstavuje skratku “Internet Software and Security Advice“ a vystihuje hlavný zámer činností, ktoré sú reakciou na postupnú integráciu informačných technológií na internetové prostredie. Spoločnosť pracuje na komerčných projektoch v oblasti vývoja a prevádzky webových systémov a poskytuje všetky s týmto súvisiace služby a kooperuje s univerzitami rovnako ako aj s VŠB-TUO. Táto firma neustále reaguje na požiadavky zákazníkov, vývoj technológií, trendov a možností v oblasti IKT. Preto vždy ponúka aktuálne komplexné zabezpečenie informačných technológií a tak isto aj riešenie marketingu a grafiky. Medzi najvýznamnejších klientov spoločnosti ISSA CZECH spadajú BAŤA a.s, Skylink S.A, Nestlé Česko s.r.o, Třinecké železářny a mnoho ďalších. Tak isto firma spolupracuje na mnohých medzinárodných vedecko-výskumných projektoch ako napríklad PREDICTUS, KNOWMAP, Automatic 3D design alebo Benchmarking portal [1].

2.1 Používané technológie vo firme

Vo firme sa nachádza množstvo pracovných pozícií zaoberajúcich sa rôznym smerom v oblasti IT. Najpoužívanejšie vývojové technológie môžeme zhrnúť v nasledujúcich bodoch:

- HTML, XHTML, CSS, XML, XSL, XSLT, AJAX, SOAP
- JAVA, PHP, .NET, vlastný framework
- Hibernate, EJB, Struts, Swing, RCP
- MySQL, MS SQL, ORACLE, MariaDB, Firebird
- CMS Designer, generátory kódov

2.2 Produkty a služby

Spoločnosť ISSA CZECH za viac ako 20 rokov pôsobenia na IT trhu ponúkla svojim klientom naozaj veľa služieb a produktov, ako napríklad internetový marketing, vývoj webových stránok a mobilných aplikácií, tvorba e-shopov, informačných systémov, dátových centier a iné.

2.2.1 Internetový marketing

Internetový marketing by sa dal popísať ako skupina činností, ktorých cieľom je získavať nových zákazníkov, udržať si ich a obchodovať s nimi, z čoho vzniká internetové podnikanie. Reklama na internete má značne nižšie náklady ako reklama v tlači alebo v médiách ako napríklad televízia. ISSA CZECH ponúka v tomto smere komplexné riešenie, ktoré začína konzultáciou so zákazníkom a určením rozpočtu. Po dohode so zákazníkom nasleduje návrh riešenia služby internetového marketingu, realizácia tejto služby, meranie účinnosti vykonávanej služby a v poslednom kroku optimalizácia na základe nameraných hodnôt, so snahou minimalizovať náklady zákazníka. Do týchto služieb môžeme zaradiť napríklad SEO optimalizáciu webových stránok, PPC kampane, banerové reklamy na internete, sociálne siete (Facebook, Twitter) a mailový marketing [2].

2.2.2 Webové stránky

Webové stránky sú vyvíjané spoločnosťou už od jej založenia, takže firma nazbierala mnoho skúseností a spĺňa aktuálne trendy v tejto oblasti. Stabilný tím grafikov a programátorov zabezpečuje prácu s veľkými ale a menšími projektami v rôznych odvetviach. Nepretržitý servis a ladenie vytvoreného webu a pravidelná optimalizácia je ďalšou prednosťou firmy. Taktiež spoločnosť disponuje vlastným redakčným systémom ISSA CMS, vďaka ktorému je v prípade potreby jednoduché meniť texty, obrázky, bannery a dokumenty na webovej stránke [3].

2.2.3 Mobilné aplikácie a E-shopy

Aplikácie na mobil podľa požiadaviek zákazníka sú tvorené pre platformy Android, Apple iOS a svojho času aj na Windows mobile. Aplikácie sú prispôbené pre chytré telefóny, tablety a môžu plniť špecifické funkcie pre širokú verejnosť ale aj pre malé firemné okruhy. Výhodou týchto aplikácií je stála mobilná dostupnosť (24/7), rýchle použitie, jednoduchá inštalácia a distribúcia [4].

Svojím zákazníkom spoločnosť ponúka tvorbu vlastného internetového obchodu podľa požiadaviek klienta. Pri vývoji využíva firma vlastné riešenie aj vlastný grafický návrh, podľa definovaných požiadaviek a funkčnej špecifikácie menších ale aj veľkých e-shopov [5]

2.2.4 Informačné systémy

Jeden z ďalších predmetov činnosti spoločnosti je vývoj softvéru pre špeciálne zákazky, ktorými je implementácia informačných systémov prostredníctvom webových technológií ako je JAVA, PHP, alebo .NET. Využívajú sa pri tom objekty, vlastné frameworky a sledovanie aktuálnych technologických trendov. Informačné systémy sú tvorené z pohľadu rôznej náročnosti, pre mnoho odvetví ako je napríklad finančníctvo, školstvo, zdravotníctvo, právnické služby a ďalšie. Na vývoji jedného informačného systému sa podieľam aj ja počas vykonávania bakalárskej praxe [6].

2.2.5 Dátové centrum

Firma riadi aj vlastné dátové centrum, ktoré zaistuje prevádzku hostingových služieb ako napríklad webhosting, emailhosting, serverhosting, FTP dátový priestor a iných služieb v rámci internetu. Základom týchto dátových centier sú hardwarové serverové jednotky, úložiská dát na viacerých lokalitách, zálohovací systém, systém rýchlej obnovy a najmä *Serverhouse*, ktorý zabezpečuje fyzické umiestnenie serverov, so záložným zdrojom elektrickej energie [7].

2.2.6 Poskytovanie internetu

Od roku 2013 ISSA poskytuje internetové pripojenie v oblasti Ostrava – Heřmanice a okolie, kde disponuje 35 metrovým telekomunikačným stožiarom, ktorý zabezpečuje kvalitné pripojenie prostredníctvom wifi pre byty a rodinné domy . Táto služba sa nazýva ISSANET a poskytované pripojenie má rýchlosť 6 Mbps. Aktivácia služby je zdarma, nemá žiadne záväzky a pripojenie je za prijateľnú cenu [8].

2.3 Popis pracovnej pozície

Pracovnú pozíciu vo firme som objavil pomocou informačného systému KatIS. Pozícia by sa dala označiť ako PHP programátor a vývojár webových aplikácií. Všetky zadania, ktoré som dostával, vychádzali z aktuálnych potrieb firmy, väčšinou týkajúce sa intranetu. Po prijatí do firmy som sa prvé dni oboznamoval s kolektívom a s pracovným prostredím. Vzhľadom k situácii som dostal možnosť vykonávať prax z domu a firmu som navštevoval za účelom odovzdávania splnenej úlohy alebo získania nového zadania. Úlohy som vykonával prevažne samostatne, a v prípade dotazov som sa obrátil na kolegu, s ktorým sme komunikovali prostredníctvom programu skype.

Kapitola 3

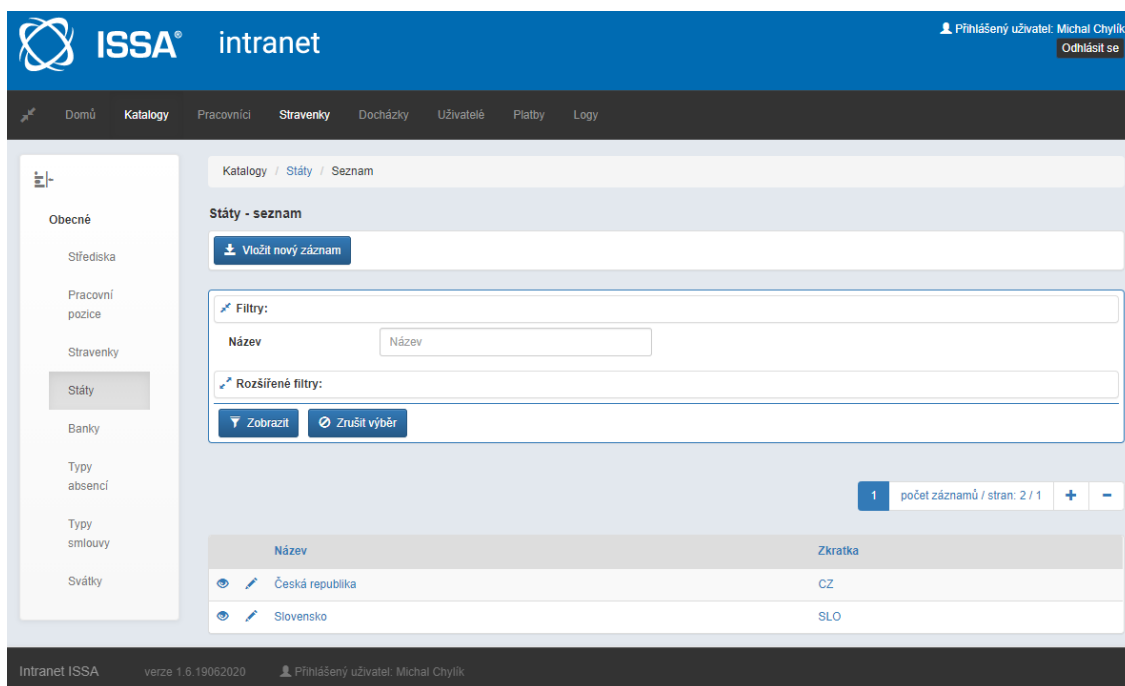
Špecifikácia zadania

V tejto kapitole by som chcel predstaviť zadanie, na ktorom som pracoval počas vykonávania bakalárskej praxe vo firme, vrátane odhadovanej časovej náročnosti jednotlivých úloh.

3.1 Firemný Intranet

Na úvod by som rád uviedol, čo predstavuje pojem Intranet. Intranet je považovaný za počítačovú sieť slúžiacu na zdieľanie informácií, operačných systémov a ďalších výpočtových služieb v rámci nejakej spoločnosti, bez prístupu osôb mimo spoločnosti. Intranet využíva rovnaké technológie ako napríklad verejná sieť internet. Taktiež pre spoločnosť predstavuje dôležité miesto pre internú komunikáciu a spoluprácu a poskytuje jediný východiskový bod pre prístup k interným a externým zdrojom. Základná podoba intranetu používa technológie pre lokálne siete (LAN) alebo rozsiahle siete (WAN). Mnoho dnešných intranetov obsahuje vo svojej infraštruktúre profily používateľov, vyhľadávače, blogy, mobilné aplikácie a plánované udalosti [9].

Intranet vo firme ISSA je určený pre potreby firmy ako napríklad administrácia zamestnancov, správa straveniek, dochádzkový systém, platby a podobne. Obsahuje mnoho užívateľov, do ktorých zapadajú všetci zamestnanci firmy na čele s adminom. Každý zo zamestnancov má právo zmeniť svoje osobné údaje, nahlasovať dovolenky, stravné lístky a zobrazovať dochádzky. Admin, na rozdiel od ostatných zamestnancov má vyššie práva a môže v systéme vykonávať kľúčové zmeny, potvrdzovať dochádzky zamestnancov a ostatné administratívne záležitosti v rámci intranetu. Hlavným dôvodom vzniku intranetu pre našu firmu bolo zjednodušenie administratívnej činnosti, ktorá bola v minulosti riešená pomocou excelovských tabuliek, čo pri rozrastaní spoločnosti spôsobovalo mnoho komplikácií.



Obr. 3.1: Intranet - stará verzia

3.2 Hlavný cieľ praxe

Cieľom celej odbornej praxe vo firme je oboznámiť sa s firemným PHP frameworkom, pomocou ktorého je nutné pretvoriť už existujúci firemný informačný systém na nový, s novými funkciami a modernejším dizajnom. Tento cieľ bol rozdelený do viacerých menších úloh. V podstate každá úloha znamenala vytvoriť nový modul informačného systému. Pre bližšiu predstavu som dostal prístup doteraz používanému systému, aby som bol oboznámený s tým, čo všetko má systém obsahovať. Moje prístupové práva boli rovnaké ako pre admina, takže som si mohol zobraziť každý modul systému, čo bol môj najväčší zdroj inšpirácie. Samotný intranet, z ktorého som čerpal kľúčové prvky jednotlivých modulov vidieť na obrázku č.1. Zadaným úlohám, ktoré som počas praxe riešil stúpala náročnosť, takže som musel často využívať samoštúdium v oblasti PHP vývoja aplikácií.

3.2.1 Modul katalógy

Odhadovaná časová náročnosť - 2 dni

Po oboznámení sa s frameworkom bolo mojou prvou úlohou vytvoriť základný modul katalógy. Tento modul obsahuje všeobecné dáta, ktoré sú neskôr využívané pri ďalších moduloch. Z pohľadu databázy je to skupina číselníkov, to znamená jednoduché tabuľky obsahujúce jeden alebo viacero atribútov viacerých dátových typov nad ktorými je možné uskutočňovať CRUD operácie. Kata-

lógy sa skladajú z tabuliek štáty, strediská, pracovné pozície, stravenky, banky, typy absencií, typy pracovnej zmluvy a sviatky. Každý číselník musí mať vytvorenú tabuľku v databáze s rovnakými atribútmi ako v starom systéme. Tak isto každá tabuľka musí obsahovať filter, niekedy aj rozšírený filter, ktorý slúži na odfiltrovanie položiek zoznamu vybrané užívateľom. Väčšinou pri týchto tabuľkách sa filtrujú záznamy podľa názvu. Výnimku obsahujú tabuľka stravenky, banky a sviatky. V neposlednom rade je nutné vytvoriť pre jednotlivé atribúty validačné práva, ktoré sa kontrolujú pri vkladaní nového záznamu do databázy. Ďalším krokom je vytvoriť samotný formulár na vkladanie dát pre každú tabuľku zvlášť. Tento formulár slúži aj ako zobrazenie už uložených záznamov, alebo umožňuje záznamy upravovať. Na záver, z dôvodu, že tabuľky a ich atribúty sú vytvorené v anglickom jazyku je potrebné napísať preklad každého z atribútov a neimplementovať na používateľské rozhranie, aby systém zobrazoval české názvy pre zjednodušenie práce so systémom.

3.2.2 Modul pracovníci

Odhadovaná časová náročnosť - 1 deň

Tento modul obsahuje zoznam všetkých zamestnancov vo firme. Tabuľka je oproti číselníkom v module katalógy omnoho rozsiahlejšia, obsahuje osobné údaje o zamestnancoch, adresu, kontaktné údaje a počty hodín, ktorý deň zamestnanec musí podľa zmluvy odpracovať. Taktiež tabuľka obsahuje počty dostupných absencií zamestnanca, ako je napríklad dovolenka. Ďalší rozdiel oproti prvému zadaniu sú nové atribúty, na ktoré museli byť aplikované špecifické validačné pravidlá. Niektoré atribúty v tabuľke sú povinné a niektoré nie, o čo sa nám postará validačné pravidlo a tak isto aj samotná vytvorená tabuľka bude môcť v týchto atribútoch nadobúdať hodnotu null. Pri vkladaní nového záznamu pomocou formulára nám systém musí poskytovať dáta z číselníkov z modulu katalógy, ako napríklad stredisko, pracovnú pozíciu alebo typ pracovnej zmluvy. Preto je nutné pri vytváraní tabuľky v databáze vytvoriť patričné cudzie kľúče do už vytvorených tabuliek. Modul treba taktiež doplniť o filter podľa mena zamestnanca a tak isto aj prekladom atribútov. Filtre a preklad sú súčasťou každej tabuľky v systéme, tak to môžeme považovať za automatickú úlohu pre každý z modulov.

3.2.3 Modul dochádzky

Odhadovaná časová náročnosť - 8 dní

Dochádzky zamestnancov sa dajú považovať za najdôležitejší a najzložitejší modul intranetu obsahujúci najviac logiky. Skladá sa z troch častí. Základom je tabuľka absencií uchováajúca informácie o jednotlivých absenciách zamestnancov. Zamestnanec si sám volí absenciu (napríklad dovolenku) niekoľko dní dopredu a čaká na schválenie adminom systému alebo nadriadeným užíva-

telom. Pri validácii formulára je potrebné vytvoriť nové validačné pravidlo, ktoré hovorí, že dátum začiatku absencie musí byť pochopiteľne menší ako dátum ukončenia absencie.

Po vytvorení tabuľky absencie je potreba vytvoriť správu dochádzok, ktorá je rozdelená na dve časti. Dochádzka jednotlivcov je tabuľka obsahujúca identifikačné číslo zamestnanca, mesiac a rok. Po kliknutí na záznam tabuľky sa nám pre konkrétného zamestnanca zobrazí záznam dochádzok z celého mesiaca vrátane všetkých jeho absencií, víkendov, sviatkov a podobne. Taktiež sa nám zobrazí zamestnancova štatistika o tom, koľko dní musel za mesiac odpracovať, koľko v skutočnosti odpracoval alebo koľko dní/hodín má ešte voľných absencií. Formulár obsahujúci tieto dáta musí byť taktiež editovateľný, v prípade nejakých zmien, ktoré treba vykonať. Dôležitou požiadavkou je vytvorenie funkcie alebo transakcie, ktorá nám vloží hromadne mesačný záznam pre každého zamestnanca. Táto funkcia je vykonávaná vždy na začiatku mesiaca za predošlý mesiac.

Posledná časť tohto modulu je súhrn dochádzok. Táto časť zobrazuje tiež dáta za určitý rok a mesiac, ale nie pre konkrétného zamestnanca ale pre všetkých zamestnancov. Tieto informácie sú zobrazené v podobe tabuľky, o veľkosti počtu aktívnych zamestnancov a počtu dní v mesiaci. Taktiež obsahuje súčet odpracovaných hodín pre každého zamestnanca, ale aj celkový mesačný súhrn všetkých odpracovaných hodín. Pre túto tabuľku je nutné vytvoriť funkciu, ktorá z vybranej mesačnej dochádzky vytvorí PDF dokument, ktorý je potrebné uchovávať pre administratívnu činnosť spoločnosti.

3.2.4 Modul stravenky

Odhadovaná časová náročnosť - 6 dní

Mojou poslednou úlohou bolo vytvoriť modul stravenky. Modul je nadviazaný na všetky predošlé moduly, preto jeho vytvorenie patrí na záver. Správa straveniek spracováva dáta tých pracovníkov, ktorí majú na svojom profile zaevidovaný príjem straveniek z modulu katalógy. Z pohľadu firmy rozdeľujeme stravenky na príjem a výdaj, čo je dôležitý faktor pri tvorení tohto modulu. Všetky položky musia byť vyčíslené a jednoznačne zobrazujú, či sa jedná o príjem alebo výdaj straveniek. Nové záznamy je možné vkladať po jednom, ale tak isto aj hromadne za celý predošlý mesiac, podobne ako v module dochádzky, s tým rozdielom že pripisujeme stravenky jednotlivým zamestnancom ručne. Kľúčovými atribútmi tabuľky straveniek je dátum vystavenia, identifikátor pracovníka, ktorý stravenky odoberá a konkrétny mesiac a rok vystavenia straveniek.

Kapitola 4

Riešenie zadaných úloh

Táto kapitola pozostáva z používaných technológií a riešenia úloh, ktoré som dostával počas bakalárskej praxe vo firme. Pri jednotlivých úlohách je uvedená ich skutočná časová náročnosť vypracovania.

4.1 Používané jazyky a technológie

Na tomto mieste popisujem používané programovacie jazyky, doplnkové knižnice a technológie, s ktorými som pracoval pri riešení zadania.

4.1.1 PHP

PHP je populárny, široko používaný open source (verejne dostupný zdroj) univerzálny programovací jazyk, ktorý sa používa najmä na programovanie klient-server aplikácií a pre vývoj dynamických webových stránok. PHP sa inšpirovalo od jazykov podporujúcich procedurálne programovanie. Najviac vlastností prebralo od jazyka C a jazyka Perl. V neskorších verziách bolo rozšírené o možnosť používať objekty. Spolupráca s relačnými databázami, ako napríklad Oracle, MySQL, Microsoft SQL Server alebo SQLite je ďalšia z výhod používania tohto jazyka. Tak isto funguje na takmer všetkých svetovo populárnych operačných systémoch ako Windows, Linux, UNIX alebo macOS [10].

Zdrojové kódy PHP sú skompilované za behu na vnútorný formát, ktorý môže byť vykonaný pomocou PHP engine. Aby sa zdrojový kód nemusel kompilovať vždy, keď pristúpi k webovej stránke, môžu byť PHP skripty taktiež nasadené v spustiteľnom formáte použitím PHP kompilera, čo urýchli čas vykonávania. Optimalizátory kódu (napr. eAccelerator, OPcache) sú určené na vylepšenie výkonu kompilácie, odstránením nadbytočných inštrukcií, redukováním veľkosti kódu a vykonávaním iných zmien, ktoré môžu znižovať vykonávací čas [11].

4.1.2 Composer

Pre zjednodušenie práce s jazykom PHP využívame Composer. Composer je multiplatformový nástroj na správu závislostí v jazyku PHP, ktorý umožňuje deklarovať a spravovať knižnice, ktoré využívame v našich projektoch. Nieje to klasický správca balíkov ako napríklad Yum, ale spravuje knižnice na základe jednotlivých projektov a inštaluje ich priamo vo vnútri projektu. Composer vyžaduje minimálne verziu PHP 5.3.2 a je funkčný na operačných systémoch Linux, Windows aj macOS [12].

4.1.3 XAMPP

Označenie XAMPP v informatike predstavuje multiplatformový softvérový balík vyvinutý firmou Apache Friends. Obsahuje voľne dostupný softvér (webový server Apache, databáza MariaDB, mail server Mercury MTS a interpret programovacích jazykov PHP a Perl). Označenie XAMPP sa skladá z piatich písmen, kde každé reprezentuje jednu spomínanú funkciu: Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) a Perl (P).

XAMPP je odľahčená distribúcia Apache, ktorá veľmi jednoducho vytvorí lokálny webový server pre testovanie a vývoj. Všetko potrebné je obsiahnuté v inštalačnom balíku - serverová aplikácia (Apache), databáza (MariaDB) a skriptovací jazyk (PHP). XAMPP je multiplatformový, čo znamená, že funguje na Linuxe, na MacOS i Windows. Vzhľadom k tomu, že väčšina implementácií webových serverov využíva rovnaké komponenty ako XAMPP, je prechod z lokálneho testovacieho servera na reálny server veľmi jednoduchý. Po inštalácii je možné sa k vzdialenému serveru pripojiť pomocou FTP klienta. Tak isto je možné sa pripojiť k miestnemu počítaču cez FTP pomocou HTML editora [13].

4.1.4 MariaDB

MariaDB Server je jedným z najrozšírenejších open source databázových serverov na svete, vytvorený pôvodnými vývojármi MySQL. Medzi najznámejších užívateľov tohto databázového serveru patrí Wikipédia, WordPress.com a Google MariaDB mení dáta na štruktúrované informácie v rozsiahlej sfére aplikácií, od bankovníctva až po webové stránky. Tento databázový server bol spočiatku navrhnutý ako lepšia náhrada za MySQL, pretože je škálovateľný a rýchly s množstvom úložných mechanizmov, doplnkov a ďalších nástrojov. Najnovšie verzie MariaDB obsahujú aj funkcie GIS a JSON [14].

4.1.5 phpMyAdmin

PhpMyAdmin najnovšej verzie 5.1.0. je softvérový nástroj písaný v jazyku PHP určený pre prácu s MySQL prostredníctvom webu. Často používané CRUD operácie alebo správu tabuliek, stĺpcov, vzťahov a indexov je možné vykonávať pomocou používateľského rozhrania, pričom je stále možné

vykonávať akýkoľvek SQL príkaz. Tento nástroj prichádza s rozsiahlou dokumentáciou, ktorú môžu používatelia aktualizovať a zdieľať tým svoje nápady a návody pre rôzne operácie.

Najväčšími výhodami je intuitívne webové rozhranie v ktorom je možné prehľadávať a sťahovať databázy, tabuľky a indexy. Taktiež podporuje importovanie dát z CSV a export do rôznych formátov ako napríklad CSV, SQL, XML, PDF a ďalšie. Umožňuje tvorbu grafického rozloženia databázy a globálne vyhľadávanie [15].

4.1.6 HTML a CSS

HTML je značkovací jazyk používaný pre návrh webových stránok. Jazyk využíva značky na definovanie manipulácie, ktorá sa vykoná s textom alebo obrázkami. Používa nedefinované značky a prvky, ktoré hovoria prehľadávaču, ako správne zobrazovať obsah. Nezabudnime používať uzatváracie značky. Ak sú vynechané, prehľadávač použije vlastnosť úvodnej značky až do konca celej stránky [16].

CSS je kaskádový štýl používaný na formátovanie rozloženia webových stránok. Taktiež sa používa na definovanie štýlov textu, farieb, veľkostí tabuliek, obrázkov a iných aspektov webových stránok, ktoré predtým definovalo len samotné HTML. Na rozdiel od definovania štýlu každého bloku samostatne v HTML stačí pomocou CSS definovať štýly len raz. To je jeden z dôvodov prečo väčšina moderných webov používa kaskádové štýly [17].

Samotné CSS je zábavné ale štýly sa rýchlo zväčšujú a komplikujú udržiavanie webu. S podobnými problémami nám pomáhajú preprocesory, a jedným z nich, ktorý je používaný taktiež v našom frameworku je Sass. Sass je jazyk štýlov, ktorý je prekompilovaný do CSS. Dáva nám možnosť používať vnorenie, premenné, mixiny, dedičnosť, a ďalšie pomocné nástroje s plne kompatibilnou syntaxou pre CSS [18].

Pre efektívnejšie štýlovanie informačného systému taktiež využívame voľne dostupný framework Bootstrap, ktorý je zameraný na front-endový responzívny webový vývoj. Obsahuje CSS šablóny pre typografiu, navigáciu, formuláre tabuľky, tlačidlá a ďalšie komponenty webovej stránky [19].

4.1.7 JavaScript

JavaScript je jeden z najznámejších, ľahko naučiteľných skriptovacích jazykov pre vývoj webových stránok, ale je používaný aj pri iných prostrediach, ktoré nepatria medzi prehliadače, ako napríklad Node.js, Apache CouchDB a Adobe Acrobat. JavaScript plne podporuje objektovo-orientované, imperatívne a deklaratívne štýly. Je spúšťaný na strane klienta na webe, čo sa využíva pri programovaní správania webových stránok pri výskyte nejakej udalosti. Základná syntax je zámerne podobná jazyku Java aj C ++, a jazykové konštrukcie, ako napríklad vetvenie (if, esle), cykly (for, while), switch a bloky try/catch fungujú rovnako ako v týchto jazykoch [20].

Pri vývoji práce s JavaScriptom je taktiež veľmi nápomocná knižnica jQuery. Pomocou tejto knižnice sa dá veľmi ľahko manipulovať s HTML dokumentom, reagovať na udalosti, vytvárať rôzne

animácie alebo asynchrónne komunikovať pomocou AJAXu. Programátorom poskytuje jednoduché API, ktoré je podporované na väčšine internetových prehliadačoch [21].

4.1.8 Twig

Twig je moderný šablónovací systém pre PHP. Réžia sa v porovnaní s bežným PHP kódom znížila na absolútne minimum. Twig taktiež dokáže vyhodnotiť nedôveryhodné kódy šablóny, čo ho umožňuje použiť ako šablónovací jazyk pre aplikácie, kde si užívatelia môžu samostatne upravovať dizajn šablóny. Syntax sa dá ľahko naučiť a je optimalizovaná tak, aby webovým dizajnérom umožnila čo najrýchlejšie dokončiť prácu [22].

4.1.9 NetBeans IDE

Pri práci s firemným frameworkom využívam bezplatné vývojové prostredie NetBeans, ktoré slúži pre vývoj aplikácií v operačných systémoch ako Windows, Linux, Mac OS, a Solaris. Toto prostredie uľahčuje vývoj webových, desktopových, podnikových, a mobilných aplikácií, ktoré využívajú platformy Java, HTML5, C/C++ alebo mnou používané PHP. Vďaka neustálemu vylepšovaniu rýchlosti a výkonu nastavuje NetBeans IDE štandard pre vývoj aplikácií s najmodernejšími technológiami. Jednoducho je možné toto IDE prepojiť s rôznymi databázovými servermi ako je Oracle, MariaDB alebo SQL server [23].

4.2 Oboznámenie sa s frameworkom

Po mojom nástupe do firmy v pozícii PHP programátora a web developera som mal možnosť zúčastniť sa mítingu, pri ktorom bola predstavená nová verzia firemného frameworku. Tento framework, písaný v jazyku PHP verzie 7.0, slúži najmä pre e-shopy a interné webové aplikácie spoločnosti. Napomáha pri evidencii záznamov pomocou základných CRUD operácií a taktiež rieši práva užívateľov systému, autentifikáciu, logy, maily, exporty dát a mnoho ďalších funkcií. Pre splnenie zadania bakalárskej praxe som sa musel s týmto frameworkom oboznámiť a naučiť pracovať. Kolega mi vysvetlil jeho základné princípy a pomohol mi prepojiť framework s vývojovým prostredím a databázou. Následne už len stačilo vytvoriť lokálny webový server pomocou programu XAMPP. Po prihlásení do systému ako admin som prvý krát videl nový dizajn intranetu.

Základom systému sú agendy užívatelia a logy, ktoré by mal obsahovať každý informačný systém. Tým pádom je možné tento systém distribuovať medzi iné spoločnosti. Mne ako vývojárovi tohto projektu, bola pridelená zložka *custom*, v ktorej som tvoril všetky ostatné moduly, práva, preklady a ostatné dokumenty, ktoré spolu tvoria celý informačný systém. Vyskúšal som si prácu s existujúcimi modulmi prostredníctvom pridania a následného zmazania užívateľa. Po zoznámení sa s frameworkom a jeho základnými funkciami som dostal prvé zadanie, ktoré sa týkalo nového modulu.

4.3 Katalógy

Skutočná časová náročnosť - 3 dni

Tento modul sa skladá z ôsmich jednoduchých tabuliek ktoré nemajú medzi sebou žiadne väzby. Rozhodol som sa začať najjednoduchšou z nich, čo bola tabuľka strediská. Obsahovala len identifikátor a názov strediska. Prvým krokom bolo nutné vytvoriť samotnú tabuľku v databáze. K tomu som použil nástroj phpMyAdmin, pomocou ktorého som do databáze jednoduchým príkazom vytvoril novú tabuľku. Po jej vytvorení som pomocou firemného frameworku vytvoril formulár, pre vkladanie nového záznamu. Následne som pridelil validačné pravidlá jednotlivým atribútom. V tomto prípade sa jednalo len o pravidlo veľkosti daného stringu pri názve strediska. Ďalším krokom bolo vytvorenie samotného zobrazenia (view) tabuľky, ktoré nám umožňuje zobrazovať detail záznamu alebo editáciu záznamu. Po tomto už len stačilo vytvoriť preklad atribútov ako už vyplýva zo zadania. Pre danú tabuľku v danom module som vytvoril súbor ktorý obsahoval preklad názvu tabuľky alebo jej atribútov. Každá z týchto pod úloh sa nachádzala v samotných triedach. Na záver bolo nutné tieto triedy nasmerovať na daný modul a vytvoriť v užívateľskom menu novú položku s názvom tejto tabuľky. Framework svojimi funkciami plne podporoval vývoj daných tried, čo mi značne uľahčovalo implementáciu tabuľky do modulu katalógy. Po úspešnom testovaní v podobe pridania záznamu a následného zmazania som pokračoval v pridávaní ostatných siedmich tabuliek. Postup som opakoval, pričom som sa stretával s ďalšími dátovými typmi pre ktoré bolo nutné pridávať rozdielne validačné pravidlá. Taktiež som vytvoril špecifický filter pri tabuľke stravenky, ktorý filtroval záznamy podľa číselnej hodnoty stravenky.

Po implementácii všetkých tabuliek do modulu, som musel tabuľky naplniť dátami zo starého intranetu. Starý intranet mi slúžil ako inšpirácia len pri pridávaní záznamov, alebo mi poskytol pohľad na logiku celého systému, nie však na použité funkcie, ktoré boli riešené pomocou starého frameworku. Pri vkladaní záznamov do tabuľky sviatky, som si uvedomil že vkladanie veľkého množstva sviatkov daného roku je príliš zdĺhavé. Z toho dôvodu som dostal dodatočné zadanie v podobe vytvorenia funkcie, ktorá vygeneruje a vloží do tabuľky všetky sviatky vo vybraný rok.

4.3.1 Generovanie sviatkov

Vytvoril som pole s pevnými dátumami, ktoré každoročne reprezentujú sviatky ako sú napríklad Vianoce. Pri veľkonočných sviatkoch, ktoré nespádajú pod rovnaký dátum každý rok, som musel zvoliť nejaký výpočtový postup. Na internete som našiel algoritmus, ktorý slúži na výpočet týchto sviatkov. Týmto algoritmom som sa inšpiroval a upravil ho podľa svojich potrieb. Z formulára som si získal rok, pre ktorý sú sviatky generované a na koniec stačilo prechádzať pole dátumov a veľkonočných sviatkov a vložiť ich postupne do databázy.

```

$year = $this->request->getAttribute('year');
$a = $year%19;
$b = $year%4;
$c = $year%7;
$d = (19*$a + 24)%30;
$e = (5 + 2*$b + 4*$c + 6*$d)%7;
$marec = 22 + $d + $e;
$april = $d + $e -9;

if($marec > 31 && $april <= 25){
    $piatok = $year.'-04-' .($april-2);
    $pondelok = $year.'-04-' .($april+1);
    if($april == 1){ $piatok = $year.'-03-30'; }
    else if($april == 2){ $piatok = $year.'-03-31'; }
}
else{
    $piatok = $year.'-03-' .($marec-2);
    $pondelok = $year.'-03-' .($marec+1);
    if($marec == 31){ $pondelok = $year.'-04-01'; }
}

$array_of_holidays = ['-01-01', '-05-01', '-05-08', '-07-05', '-07-06',
    '-09-28', '-10-28', '-11-17', '-12-24', '-12-25', '-12-26'];
$cis_holiday = new CisHoliday($this->dbm);
$cis_holiday->date = $pondelok;
$cis_holiday->save();
$cis_holiday = new CisHoliday($this->dbm);
$cis_holiday->date = $piatok;
$cis_holiday->save();

foreach ($array_of_holidays as $date) {
    $cis_holiday = new CisHoliday($this->dbm);
    $cis_holiday->date = $year.$date;
    $cis_holiday->save();
}

```

Výpis 4.1: Funkcia generovania sviatkov

4.4 Pracovníci

Skutočná časová náročnosť - 2 dni

Modul pracovníci slúži k evidencii všetkých zamestnancov firmy. Pozostáva len z jednej tabuľky, ktorá má viac než 30 atribútov a tým sa stáva najrozsiahlejšia v celom intranete. Postup vytvorenia tohto modulu bol podobný ako u všetkých katalógov, s tým rozdielom, že tabuľka obsahovala atribúty, ktoré predstavovali cudzie kľúče do iných tabuliek. Po vytvorení tabuľky bolo nutné vytvoriť relácie do ostatných tabuliek z modulu katalógy. Relácie sa viazali na tabuľky užívateľ, stravenky, strediská, pracovná pozícia, typ pracovnej zmluvy, banky a štáty.

```
ALTER TABLE 'empl_employee'
ADD CONSTRAINT 'fk_eem_au' FOREIGN KEY ('id_ar_user')
REFERENCES 'ar_user' ('id') ON DELETE CASCADE ON UPDATE CASCADE;
ADD CONSTRAINT 'fk_eem_cc' FOREIGN KEY ('id_cis_center')
REFERENCES 'cis_center' ('id') ON DELETE CASCADE ON UPDATE CASCADE;
ADD CONSTRAINT 'fk_eem_cs' FOREIGN KEY ('id_cis_state')
REFERENCES 'cis_state' ('id') ON DELETE CASCADE ON UPDATE CASCADE;
ADD CONSTRAINT 'fk_eem_cmt' FOREIGN KEY ('id_cis_meal_ticket')
REFERENCES 'cis_meal_ticket' ('id') ON DELETE SET NULL ON UPDATE CASCADE;
ADD CONSTRAINT 'fk_eem_cb' FOREIGN KEY ('id_cis_bank')
REFERENCES 'cis_bank' ('id') ON DELETE SET NULL ON UPDATE CASCADE;
ADD CONSTRAINT 'fk_eem_cct' FOREIGN KEY ('id_cis_contract_type')
REFERENCES 'cis_contract_type' ('id') ON DELETE CASCADE ON UPDATE CASCADE;
ADD CONSTRAINT 'fk_eem_cj' FOREIGN KEY ('id_cis_job')
REFERENCES 'cis_job' ('id') ON DELETE CASCADE ON UPDATE CASCADE;
```

Výpis 4.2: Relácie tabuľky pracovníci

Vytvorené relácie bolo potrebné pomocou funkcií definovať aj v samotnom module a prepojiť tým všetky potrebné tabuľky. Vo formulári stačilo už len na samotné vstupy zavolať tieto funkcie, ktoré nám pomocou drop-down listu poskytovali výber dát z danej tabuľky ako napríklad typ pracovnej zmluvy alebo pracovnú pozíciu. Filter v tomto module filtroval záznamy len podľa mena a priezviska zamestnanca. Každému atribútu som pridal validačné pravidlo, ktoré muselo byť splnené pri vkladaní nového záznamu do tabuľky. Napokon zostával len preklad všetkých atribútov do českého jazyka rovnako ako pri katalógoch.

4.5 Dochádzky

Skutočná časová náročnosť - 14 dní

Tento modul som započal tvorbou tabuľky absencie, ktorá pozostáva s dátumu a času vloženia záznamu užívateľa, ktorý absenciu vložil, zamestnanca, pre ktorého platí daná absencia, časové rozmedzie od kedy do kedy a typ absencie ako napríklad dovolenka. Tabuľka obsahuje aj atribút typu boolean, ktorý hovorí o tom, či bola daná absencia schválená administrátorom alebo inou nadriadenou osobou vrátane dátumu schválenia a osoby, ktorá absenciu schválila. Taktiež je potreba ukladať údaje o tom, či sa do absencie zarátavajú víkendy, sviatky a dni voľna. Doposiaľ boli všetky absencie uchovávané v dňoch, no pri vytváraní tohto modulu som na žiadosť firmy musel všetky absencie riešiť v hodinách. To sa odvíjalo od toho, koľko hodín má každý zamestnanec pridelených pracovným dňom. V poslednom rade bolo možné vložiť poznámku ku každému záznamu.

Po vytvorení tabuľky som musel znova vytvoriť relácie medzi ostatnými tabuľkami a to konkrétne zamestnanci, užívatelia a typ absencie. Postup tvorenia celej agendy absencií bol podobný ako pri predošlých moduloch, čo znamenalo vytvorenie formulára pre vkladanie záznamov, filter a rozširujúci filter. Pri zobrazení tabuľky v systéme, kde zobrazujeme len vybrané atribúty prišla malá zmena. Tabuľka zobrazovala len identifikátory atribútov a nie celé názvy ako by bolo pre používateľa vhodné. V kontroléri tohto modulu som musel vytvoriť funkciu s sql dotazom, ktorý mi vracal potrebné údaje a až tie boli zobrazované v používateľskom rozhraní. Pri validácii funkčnosť systému vyžadovala vytvorenie nového pravidla, ktoré nebolo medzi existujúcimi pravidlami vo firmnom frameworku. Pravidlo riešilo dátumy absencie, pri ktorých nemohol byť dátum začiatku absencie pochopiteľne väčší ako jej koniec. Inšpiráciu som hľadal medzi ostatnými pravidlami, ktoré fungovali na podobnom princípe, kde sa do parametru pravidla odosieli 2 atribúty a následne boli porovnané podľa potreby. Novo vytvorené pravidlo som nazval „LessThenRule”, aby som dodržal isté zásady frameworku. Napokon už stačilo priradiť atribútom jednotlivé pravidlá a vytvoriť preklad všetkých atribútov a názvu tabuľky, tak ako doteraz pri ostatných moduloch.

4.5.1 Dochádzky jednotlivca

Táto agenda pozostáva zo strany databázy len z jednoduchej tabuľky, ktorá zahŕňa atribúty ako identifikátor zamestnanca, mesiac a rok dochádzky. Ako pri ostatných agendách som vytvoril tabuľku, filtre, formulár, validáciu a preklad do českého jazyka. Rozdiel však prišiel pri zobrazení detailu záznamu. Tu som sa prvý krát stretol s vytváraním vlastného view pomocou HTML a šablónovacieho systému Twig. Vytvoril som prázdny súbor s príponou „html.twig” a nalinkoval som ho do formulára detailu záznamu na miesto pod menom zamestnanca, rokom a mesiacom danej dochádzky. Tento vlastný pohľad (view) pozostával z dvoch tabuliek.

Prvá tabuľka zobrazovala prehľad celého mesiaca, odpracované hodiny v každom dni a všetky absencie uskutočnené v tomto mesiaci. Druhá tabuľka obsahovala zamestnancovú štatistiku absencií v priebehu daného mesiaca a prípadne počet zvyšných dní dovolenky. Po zoznámení sa, so šablónovacím systémom nebol problém tabuľku vytvoriť, avšak postup spracovania dát už nebol tak intuitívny. Všetky výpočtové funkcie, ktoré som pri tvorení tabuľky využil, boli implementované v triede so samotným formulárom. Potom potrebné dáta boli odoslané ako parameter funkcie pre zobrazenie tohto view.

Základom bolo spracovať konkrétny mesiac vo vybranom roku a zistiť počet jeho dní podľa gregoriánskeho kalendára. Tieto získané dátumy som si uložil do poľa a odoslal ich do view, kde som si ich pomocou jednoduchého for cyklu vložil do tabuľky. Ku každému dátumu som patrične pridal aj informáciu o tom, o aký deň v týždni sa jedná. Táto informácia sa dala jednoducho vybrať z daného dátumu pomocou PHP funkcií pracujúcich s objektmi typu date. Následne som pridal preklad názvov dní, aby boli v tabuľke zobrazované ako Po, Ut, St ... Všetky riadky v tabuľke, ktoré reprezentovali soboty, nedele, sviatky a dni voľna, museli byť farebne alebo inak odlišené od ostatných. Víkendy bolo jednoduché oddeliť od pracovných dní pomocou vetvenia IF, avšak som musel vytvoriť vlastnú funkciu, ktorá mi vráti pracovné dni v danom mesiaci. Sviatky a ostatné dni voľna som musel dostať z tabuľky sviatkov z modulu katalógy.

Ďalším krokom bolo dostať z tabuľky absencií celú kolekciu zamestnancových absencií pre vybraný mesiac. Z tejto kolekcie som si vybral potrebné dáta ako je dátum začiatku a konca absencie, typ absencie a poznámku. Taktiež som tieto informácie uložil do poľa a dátumy kde nebola žiadna absencia, obsahovali hodnotu null. Pomocou ďalšieho for cyklu som tieto dáta vložil do tabuľky mesačnej absencie. Časť vyplnenej tabuľky absencií môžeme vidieť na obrázku č. 4.1.

The screenshot shows a web application interface for managing employee attendance. The sidebar on the left contains a menu with items like Dashboard, Katalogy, Pracovníci, Docházky, Stravenky, and Uživatelé. The main content area displays the 'Docházky jednotlivci' (Individual Attendance) section for the employee 'Michal Chylík' in the year '2021' and month '3' (March). A table shows the attendance data for each day of the month, with columns for 'Den' (Day), 'Datum' (Date), 'Odpracováno' (Hours worked), 'Absence' (Absence), and 'Poznámka' (Note). The table highlights weekends (Saturday and Sunday) in red.

Den	Datum	Odpracováno	Absence	Poznámka
Po	1.03.2021	4		
Út	2.03.2021	4		
St	3.03.2021		A	
Čt	4.03.2021		A	
Pá	5.03.2021		A	
So	6.03.2021			
Ne	7.03.2021			
Po	8.03.2021	4		

Obr. 4.1: Dochádzky jednotlivca

Štatistika mesačných absencií fungovala, podobne. Vytvoril som pole o veľkosti počtu typov absencií a do každej absencie zapísal jej počet vyskytujúci sa vo vybranom mesiaci. Tieto dáta som vložil do tabuľky štatistiky, ktorá vyzerala nasledovne na obrázku č. 4.2.

Plánovano	23 dní / 92 hodín
Odpracováno	13 dní / 52 hodín
Nahradní Volno	0
Nemocenská	0
Neomluvená absence	3
Neplacené volno	8
Paragraf k proplacení	0
Služební cesta	0
Indispoziční volno (sick days)	0 / 4 dní zbýva
Dovolená	0 / 20 hodín zbýva
Otcovská dovolená	0

Obr. 4.2: Štatistika jednotlivca

Celý mesačný záznam absencie zamestnanca musel byť editovateľný. Preto som vytvoril nové view, ktoré bolo takmer rovnaké ako vyššie popísané, ale namiesto vypísaných záznamov, boli bunky tabuľky zobrazované ako input boxy pre zmenu údajov. Celá tabuľka bola obalená vo formulári, ktorý po stlačení tlačidla *editovať* uloží dáta do novo vytvorenej tabuľky v databáze, obsahujúcej identifikátor, dátum, a počet odpracovaných hodín. Identifikátor tabuľky bude zároveň cudzím kľúčom do tabuľky absencií jednotlivca. Tieto dáta musia byť uložené v externej tabuľke z dôvodu editácie a následného využitia pri ďalšej agende mesačného súhrnu dochádzok.

4.5.2 Dochádzka mesiaca

Posledná agenda modulu dochádzok zobrazuje informácie o absencií nielen jedného zamestnanca, ale všetkých aktívnych zamestnancov firmy a tak isto štatistiku za celý mesiac. Vytvoril som vlastné view, ktoré obsahovalo jednu veľkú tabuľku pre všetky spomínané dáta. Dáta som získaval podobne ako pri predošlej agende a mesačné absencie jednotlivých zamestnancov mi vracala novo vytvorená databázová tabuľka absencií jednotlivca. Všetky dáta som mal uložené v dvojrozmernom poli o veľkosti počtu zamestnancov a dĺžky mesiaca a následne som ich pomocou cyklov vkladal do tabuľky.

Táto agenda na rozdiel od ostatných už neobsahovala možnú editáciu záznamov, ale za to obsahuje funkciu, ktorá vygeneruje z celej dochádzky PDF dokument. Ako vzor dokumentu som si vygeneroval záznam zo starého intranetu, podľa ktorého som vytvoril nový HTML súbor. Tabuľky v tomto súbore som naplnil dátami rovnako ako absencie mesiaca. Pomocou PHP Composera som si nainštaloval knižnicu s názvom *HTML2PDF*, ktorá slúži na tvorbu PDF dokumentu zo sú-

borov iného formátu. Stačilo už len vytvoriť funkciu, ktorá pomocou spomínanej knižnice vytvorí z HTML súboru PDF dokument a bude zavolaná pri stlačení na tlačidlo *Generovať PDF*. Vygenerované PDF sa automaticky stiahne do zariadenia. Ukážka vygenerovaného PDF dokumentu sa nachádza v prílohe tejto práce.

4.6 Stravenky

Skutočná časová náročnosť - 9 dní

Postup pri tvorbe tohto modulu, slúžiaceho na prideľovanie stravných lístkov zamestnancom bol podobný ako pri doterajších moduloch. Tvorba tabuľky straveniek, obsahujúca 10 atribútov, bola nasledovaná vytvorením relácií medzi ďalšími potrebnými tabuľkami. V našom prípade sme pripojili tabuľku pracovníkov a straveniek z katalógov. Na to nadväzovalo vytvorenie formulára, validácia, a samozrejme vytvorenie základného a rozšíreného filtra.

```
CREATE TABLE IF NOT EXISTS 'strav_mealticket' (  
  'id' int(11) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT 'ID',  
  'date' date NOT NULL COMMENT 'dátum',  
  'id_cis_meal_ticket' int(11) UNSIGNED NOT NULL COMMENT 'typ stravenky',  
  'pieces' int(11) UNSIGNED NOT NULL COMMENT 'počet kusu',  
  'use' BOOLEAN NOT NULL COMMENT 'účel',  
  'id_empl_employee' int(11) UNSIGNED COMMENT 'pracovník',  
  'strav_year' int(11) UNSIGNED COMMENT 'rok stravného',  
  'strav_month' int(11) UNSIGNED COMMENT 'mesíc stravného',  
  'confirmation' BOOLEAN COMMENT 'schváleno',  
  'value' int(11) COMMENT 'hodnota stravenek',  
  'note' varchar(500) COMMENT 'poznámka',  
  PRIMARY KEY ('id')  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_czech_ci COMMENT='  
  Stravenky';
```

Výpis 4.3: Štruktúra tabuľky stravenky

Pri zobrazení tabuľka zobrazovala len číselné hodnoty z ostatných tabuliek, preto som musel dostať ich korektné názvy a implementovať ich na požadované miesto v tabuľke. Taktiež bolo potrebné zobraziť celkovú finančnú hodnotu každého záznamu a farebné odlíšenie tých záznamov, ktoré ešte neboli potvrdené administrátorom. K tomu stačilo priamo pristúpiť k HTML kódu danej tabuľky a upraviť potrebné údaje a vytvoriť štýl pre farebné odlíšenie záznamov.

Pri tvorbe formulára som musel prvý krát využiť skriptovací jazyk JavaScript. Ak užívateľ systému pri tvorbe nového záznamu zvolí, že sa jedná o príjem straveniek, zablokuje to ostatné vstupy pracovníka, rok a mesiac stravenky. To som docielil pomocou knižnice jQuery, kde som si nadefinoval funkciu na radio box s hodnotou príjem a po stlačení nastavilo požadované vstupy na stav disabled (zakázaný). Na opak po stlačení radio boxu s hodnotou výdaj, opäť nastavilo vybrané vstupy na stav enabled (povolený).

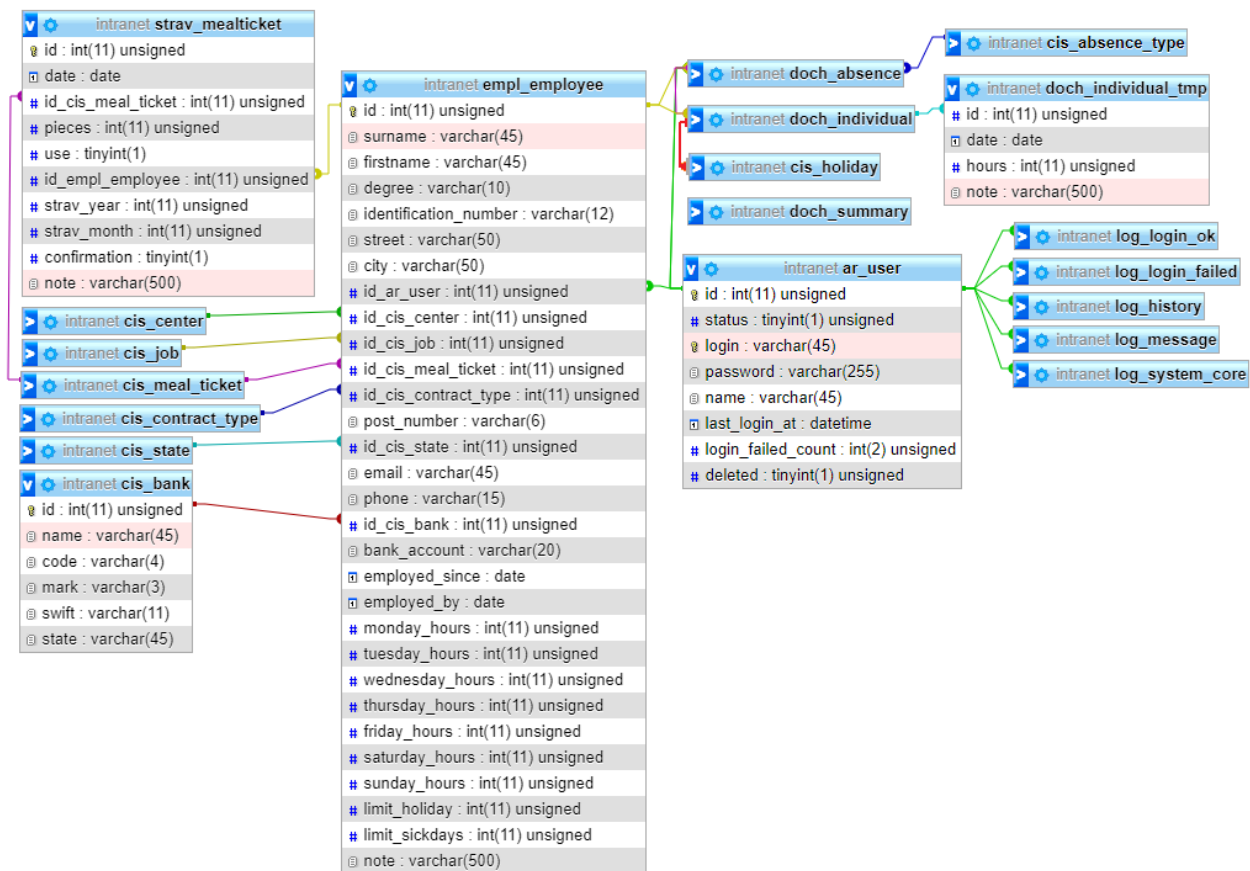
Obr. 4.3: Formulár modulu stravenky

Mojou poslednou úlohou v module stravenky a zároveň poslednou úlohou celej bakalárskej praxe bolo vytvoriť formulár pre hromadné vloženie straveniek všetkým vybraným zamestnancom. Vytvoril som si nový HTML dokument obsahujú formulár s menom zamestnanca, mesiac a rok a typ stravenky a vstupný box pre zadanie počtu kusov.

V kontroléri modulu stravenky som vytvoril funkciu, do ktorej boli zaslané vstupné parametre rok a mesiac. Následne som si z databázy vytiahol kolekciu všetkých zamestnancov a odfiltroval len tých ktorí majú stav aktívny a odoberajú stravenky. Týchto vybraných zamestnancov som si uložil do poľa, ktoré bolo odoslané do vytvoreného formulára. Napokon už stačilo len pridať tlačidlo na odoslanie formulára a priradiť mu funkciu hromadného vloženia straveniek.

4.6.1 Databáza

V tejto časti bakalárskej práce by som viac priblížil pohľad na databázu celého informačného systému. Jednotlivé tabuľky, ako už bolo spomenuté, boli tvorené pri vytváraní samotných modulov. Na obrázku 4.4 môžeme vidieť celú databázu, vrátane všetkých prepojení tabuliek na základe cudzích kľúčov. Niektoré z hlavných tabuliek ako napríklad *pracovníci* zobrazujem detailne, aby bolo možné vidieť všetky podstatné atribúty, ich dátové typy a najmä relácie, ktoré zabezpečujú funkčnosť systému. Tabuľky, ktoré sa starajú o logy užívateľov už boli vytvorené pred odbornou praxou, ale musel som ich zahrnúť aby bola databáza kompletná. Obrázok bol vytvorený pomocou nástroja phpMyAdmin, ktorý obsahuje aj zobrazovaný dizajnér databáz.



Obr. 4.4: Návrh databázy intranet

4.7 Moje hodnotenie

Práca na tomto projekte ma bavila a bola pre mňa obrovskou skúsenosťou do života. Značne som zlepšil svoje programátorské znalosti a naučil sa pracovať s novými technológiami. Osobne si myslím, že som zadanie splnil, ale verím, že sa mnoho vecí dalo vypracovať efektívnejšie, s menšou algoritmickou zložitou. Najväčšou oporou mi bol samotný framework, ktorý mi svojimi funkciami bol nápomocný v každom smere. Čo sa týka časovej náročnosti všetkých zadaní, bola primeraná k môjmu času popri štúdiu. Dovolím si prehlásiť, že teraz by mi práca na firemnom intranete zabrala kratší čas z dôvodu, že už ovládam prácu s firemným frameworkom na vyššej úrovni ako bolo na začiatku bakalárskej praxe. Samozrejme, ako býva pri práci programátora zvykom, odhadovaná časová náročnosť sa líšila od tej skutočnej. V niektorých prípadoch až dvojnásobne. Každopádne, najviac času trvalo samotné testovanie funkčnosti systému a následná oprava chýb pri každom module. V nasledujúcej tabuľke porovnávam tieto časové náročnosti.

Modul	Odhadovaná náročnosť (dni)	Skutočná náročnosť (dni)
Katalogy	2	3
Pracovníci	1	2
Dochádzky	8	14
Stravenky	6	9
Oprava chýb	-	15

Tabuľka 4.1: Časová náročnosť zadaných úloh

4.7.1 Uplatnené znalosti získané počas štúdia

V tejto sekcii by som rád popísal všetky znalosti a skúsenosti, ktoré mi prinieslo doterajšie štúdium na vysokej škole, a ktoré som uplatnil počas vykonávania odbornej praxe vo firme. Budem vychádzať s konkrétnych predmetov a kurzov ktoré ma obohatili o tieto znalosti.

Na začiatku odbornej praxe pri oboznamovaní sa s frameworkom mi značne pomohla znalosť objektovo orientovaného programovania, s ktorým sme sa naučili pracovať na predmete *Programovanie II*. Samozrejme najväčšou podporou pri programovaní boli všeobecné znalosti z predmetov *Algoritmy I*, *Algoritmy II* a *Programovanie I*, ktoré mi rozšírili moje základné znalosti nadobudnuté počas štúdia na strednej škole.

Ďalej by som uviedol znalosti z kurzov *Úvod do databázových systémov* a *Databázové a informačné systémy*. Tieto kurzy mi veľmi pomohli pri práci s databázou, s ktorou som pracoval počas celej praxe. Naučili ma tvoriť jednoduché aj zložitejšie SQL dotazy. Taktiež som sa naučil pracovať s návrhom databázy, tvorbou tabuliek a rôznymi dátovými typmi, ktoré som používal aj pri databáze firemného Intranetu.

Užitočné znalosti, ktoré som využil najmä pri vývoji front-endovej časti intranetu som získal v predmete *Vývoj internetových aplikácií*, ktorého som sa účastnil počas vykonávania praxe. Tu som tak isto rozšíril moje znalosti v oblasti jazykov tvoriacich web ako je napríklad HTML, CSS a skriptovací jazyk JavaScript, pomocou ktorého som riešil úpravu vstupov niektorých formulárov v systéme. Predmet mi otvoril oči aj v oblasti moderných trendov a dizajnu webových stránok.

Nemal by som zabudnúť spomenúť kurz *Vývoj informačných systémov*, pretože som s informačným systémom pracoval a utvrdil si tým získané znalosti z tohto predmetu. Tu som získal ucelený pohľad na informačné systémy, akým princípom fungujú a aká je ich podstata. Detailný vývoj systému mi bol priblížený v kurze *Úvod do softwarového inžinierstva*, kde som sa naučil že dobre navrhnutý systém je polovica úspechu.

4.7.2 Chýbajúce znalosti v priebehu praxe

Na tomto mieste uvediem znalosti, ktoré mi počas vykonávania odbornej praxe chýbali a musel som ich dopĺňať samoštúdiom.

V prvom rade mi chýbali značné znalosti a skúsenosti s jazykom PHP, s ktorým som sa síce stretol počas štúdia na strednej škole, ale to bol len hrubý základ. Zblížiť sa týmto jazykom nepredstavovalo až tak veľký problém keďže jeho syntax a kľúčové slová sú dosť podobné jazyku C. Vetvenie *if*, cykly *for* a *while* a návratové hodnoty funkcií sú syntakticky veľmi blízke jazykom C++, C# alebo Java, ktoré ma sprevádzali počas celého štúdia.

Napriek tomu že som absolvoval predmet *Vývoj internetových aplikácií*, chýbali mi určité znalosti v oblasti HTML a JavaScriptu, pretože predmet nezachádzal do hlbšej problematiky, ktorú som musel riešiť pri vývoji intranetu. Určite by mi pomohlo zaradenie jazyka PHP do školskej osnovy aspoň v takom množstve ako JavaScript.

Nedostatky som pocítil aj pri stretnutí s databázou MariaDB, avšak bohaté znalosti z ostatných databázových systémov ako je Microsoft SQL server alebo Oracle neboli tak odlišné. S malými syntaktickými rozdielmi a inými odlišnosťami som sa oboznámil pomocou samoštúdia za pomerne krátky čas.

Určite by som privítal väčšiu znalosť v oblasti systémov správy verzií. Zo základmi verzovacích systémov (konkrétne GIT), som sa stretol na predmete *Tvorba mobilných aplikácií*, no ostatné potrebné znalosti som získaval vo voľnom čase.

Mojím najväčším nedostatkom a zároveň oporou bol samotný firemný framework, kde mi spočiatku chýbalo veľmi veľa skúseností ale časom som sa s ním naučil pracovať a vo veľkom rozsahu využíval jeho funkcionality.

4.7.3 Získané znalosti počas vykonávania odbornej praxe

Počas vykonávania odbornej praxe v spoločnosti ISSA CZECH, som okrem mnoho skúseností z pracovného prostredia získal aj množstvo znalostí v oblasti informačných a komunikačných technológií. Zdokonalil som svoje základné znalosti vývoja webových stránok, čo sa týka jazyka HTML, CSS, JavaScriptu s knižnicou jQuery a tak isto front-endového frameworku Bootstrap.

Najväčším prínosom boli znalosti jazyka PHP, ktoré som získal najmä pri práci s firemným frameworkom a pomocou samoštúdia. Naučil som sa ako jazyk PHP využiť vo všeobecnosti v rámci internetových aplikácií, a ako s týmto jazykom pracujú skúsení programátori. Počas môjho pôsobenia v spoločnosti som sa stretával pravidelne s čítaním a spracovaním cudzieho kódu, čo tiež považujem za dôležitú schopnosť programátora.

V neposlednom rade som si vyskúšal prácu z databázou MariaDB, čo bolo pre mňa novinkou, a práca s nástrojom phpMyAdmin. Pomocou MySQL syntaxe som tvoril medzi tabuľkami netradičné relácie, s ktorými som sa počas štúdia nestretol.

Veľa som sa naučil aj o verzovacích systémoch, pretože som pravidelne s kolegami vo firme riešil aktualizáciu najnovšej verzie PHP frameworku. K tomu bol využívaný program Fork, pomocou ktorého som aj ja mohol mať vytvorenú vlastnú vetvu a zdieľať svoju vypracovanú časť projektu.

Celá odborná prax mi vo všeobecnosti rozvinula komunikačné znalosti, tímovú prácu v kolektíve, zvýšenie efektivity samoštúdia a najmä stať sa zodpovedným človekom, pretože to je veľmi dôležitá vlastnosť zamestnanca v každej spoločnosti.

Kapitola 5

Závěr

Celú odbornú prax, ktorú som absolvoval hodnotím veľmi pozitívne a som rád že naša univerzita dáva študentom možnosť vykonávať odborné praxe vo firmách. Vedenie aj kolegovia zo spoločnosti ISSA CZECH mali dostatok trpezlivosti pri mojich študentských skúsenostiach s používanými technológiami a boli veľmi nápomocný, keď som potreboval poradiť alebo niečo vysvetliť, hlavne pri používaní firemného PHP frameworku.

Túto možnosť odbornej praxe som využil, pretože som nemal žiadne pracovné skúsenosti v oblasti informačných technológií sa chcel som si vyskúšať pozíciu programátora, ktorej výstupom bude nejaký systém alebo aplikácia ktorá bude denne používaná či už medzi bežnými ľuďmi, alebo v menšom okruhu nejakej spoločnosti.

Som veľmi vďačný že ma spoločnosť ISSA CZECH prijala medzi svoj kolektív a obdarovala ma mnohými znalosťami a skúsenosťami. Naučil som sa pracovať s novými zaujímavými technológiami a tak isto som sa zdokonalil v programovacích jazykoch ktoré som už predtým poznal.

Celkovým výstupom mojej celoročnej odbornej praxe je informačný systém pre správu dát vo firme v novom dizajne s novými funkciami založený na firemnom PHP frameworku. Systém slúži na hlásenie absencií zamestnancov a ich potvrdzovanie, prideľovanie stravných lístkov zamestnancom, listovanie katalógov a editáciu osobných údajov. Spoločnosť mi ponúkla možnosť ďalšej spolupráce, pri ktorej bude firemný intranet ešte viac rozširovaný. V tejto spolupráci budem rád pokračovať popri štúdiu v ďalších ročníkoch.

Literatúra

1. *ISSA.cz* [online] [cit. 2021-02-21]. Dostupné z : <https://www.issa.cz/o-nas-620.html>.
2. *Marketing* [online] [cit. 2021-04-12]. Dostupné z : <https://www.issa.cz/internetovy-marketing-622.html>.
3. *Webové stránky* [online] [cit. 2021-04-12]. Dostupné z : <https://www.issa.cz/webove-stranky-637.html>.
4. *Mobilné aplikácie* [online] [cit. 2021-04-12]. Dostupné z : <https://www.issa.cz/mobilni-aplikace-631.html>.
5. *E-shopy* [online] [cit. 2021-04-12]. Dostupné z : <https://www.issa.cz/e-shopy-624.html>.
6. *Informačné systémy* [online] [cit. 2021-04-12]. Dostupné z : <https://www.issa.cz/informacni-systemy-626.html>.
7. *Datové centrum* [online] [cit. 2021-04-12]. Dostupné z : <https://www.issa.cz/datove-centrum-625.html>.
8. *Internet* [online] [cit. 2021-04-12]. Dostupné z : <https://www.issa.cz/poskytovani-internetu-ostrava-issanet-35.html>.
9. *Intranet* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-03-10]. Dostupné z : <https://en.wikipedia.org/wiki/Intranet>.
10. *Php* [online] [cit. 2021-03-04]. Dostupné z : <https://www.php.net/docs.php>.
11. *Php* [online] [cit. 2021-03-04]. Dostupné z : <https://www.php.net/manual/en/intro-whatcando.php>.
12. *Composer* [online] [cit. 2021-02-21]. Dostupné z : <https://getcomposer.org/doc/00-intro.md>.
13. *XAMPP* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-02-21]. Dostupné z : <https://cs.wikipedia.org/wiki/XAMPP>.
14. *MariaDB* [online] [cit. 2021-02-21]. Dostupné z : <https://mariadb.org/about/>.
15. *PhpMyAdmin* [online] [cit. 2021-03-20]. Dostupné z : <https://www.phpmyadmin.net/>.

16. *HTML* [online] [cit. 2021-03-04]. Dostupné z : <https://www.geeksforgeeks.org/html-introduction/>.
17. *Kaskadové-štýly* [online] [cit. 2021-03-05]. Dostupné z : <https://techterms.com/definition/css>.
18. *Sass* [online] [cit. 2021-03-05]. Dostupné z : <https://sass-lang.com/guide>.
19. *Bootstrap* [online] [cit. 2021-03-05]. Dostupné z : [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)).
20. *JavaScript* [online] [cit. 2021-04-06]. Dostupné z : https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript.
21. *jQuery library* [online] [cit. 2021-04-12]. Dostupné z : <https://api.jquery.com/>.
22. *Twig* [online] [cit. 2021-03-05]. Dostupné z : <https://twig.symfony.com/doc/3.x/intro.html>.
23. *NetBeans: IDE* [online] [cit. 2021-02-21]. Dostupné z : <https://www.oracle.com/tools/technologies/netbeans-ide.html>.

Dodatok A

Dochádzky súhrn

Táto príloha obsahuje tlačový výpis mesačnej dochádzky zamestnancov spoločnosti, ktorý je generovaný pomocou webovej aplikácie intranetu.

Název soustavy: Docházky souhrn

Měsíc a rok: 3/2021

Dátum a čas tisku: 29.4.2021 23:30:02

Každá směna automaticky obsahuje přestávku, dle zákoníku práce, po 4,5 hodinách od započetí pracovní doby.

Počet odpracovaných hodin

NV: Náhradní volno, N: Nemocenská, A: Neomluvená absence, NE: Neplacené volno, P: Paragraf k proplacení, SC: Služební cesta, IV: Indispoziční volno (sick days), D: Dovolená, OD: Otcovská dovolená

Jméno a příjmení	Po 1	Út 2	St 3	Čt 4	Pá 5	So 6	Ne 7	Po 8	Út 9	St 10	Čt 11	Pá 12	So 13	Ne 14	Po 15	Út 16	St 17	Čt 18	Pá 19	So 20	Ne 21	Po 22	Út 23	St 24	Čt 25	Pá 26	So 27	Ne 28	Po 29	Út 30	St 31
Anonymná osoba 1	4	4	A	A	A	A		4	4	4	4	4			4	4	NE	NE	NE	NE	NE	NE	NE	NE	NE	4			4	4	4
Anonymná osoba 2	8	8	8	8	8			8	8	8	8	8			8	8	8	8	8			8	8	8	8	8			8	8	8
Anonymná osoba 3	6	6	6	6	6			6	6	6	6	6			NV	NV	NV	6	6			6	6	6	6	6			6	6	6
Anonymná osoba 4	D	D	D	6	6			6	6	P	P	P	P		6	6	6	6	6			6	6	6	6	6			6	6	6

Jméno a příjmení	NV	N	A	NE	P	SC	IV	D	OD	Celkem hodin
Anonymná osoba 1	0	0	4	9	0	0	0	0	0	136
Anonymná osoba 2	0	0	0	0	0	0	0	0	0	136
Anonymná osoba 3	3	0	0	0	0	0	0	0	0	136
Anonymná osoba 4	0	0	0	0	4	0	0	3	0	136

Počet záznamů: 4